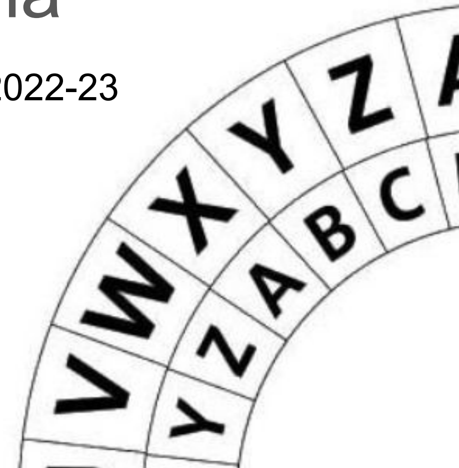
Text Decryption

Michele Alessi, Samuele D’Avenia

Università degli studi di Trieste: Probabilistic Machine Learning A.A. 2022-23



Introduction

mggz qwdlfbggb
dgzqv il ijcc kogi
qb qnncjsqdjgb gw
ojzzlb pqfrgy
pgzlck wgf
zlsfvndjgb

good afternoon
today we will
show ...

mggz qwdlfbggb
dgzqv il ijcc
kogi ...

good afternoon mggz qwdlfbggb
today we will dgzqv il ijcc
show ... kogi ...

Substitution cipher: random permutation of the alphabet.
(26! permutations)

A	B	C	...	O	...	W	...	Z
|	|	|		|		|		|
Q	H	S	...	G	...	I	...	E

```
good afternoon          dx6l pyc7zn6x8
today we will           36lph kj kgrr
show ...                ot6k
```

Homophonic cipher:
Assigns extra symbols as well (in this case numbers)

A	B	C	...	O	...	W	...	Z
P	M	W	...	X	...	K	...	S
				6				0

Bigram probabilities

‘e ‘	0.03585750
‘ch’	0.03353880
‘b ’	0.00028319
‘ao‘	0.00001431

Count occurrences of all possible bigrams in the training text and divide by the total number of occurrences.

MCMC exploration [1]

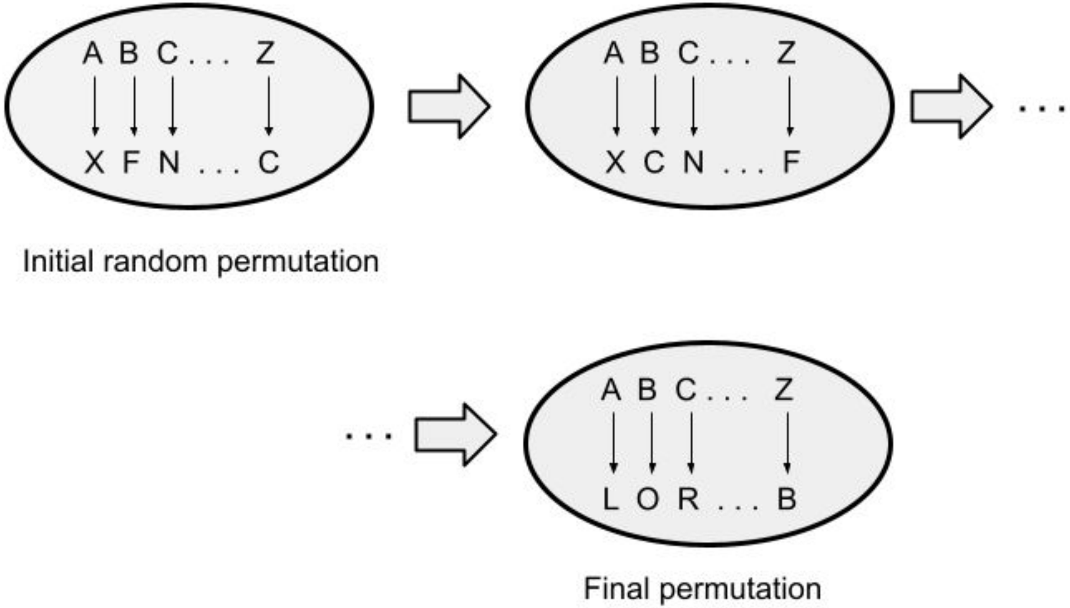
The likelihood of a certain phrase is given as:

$$\mathbf{L} = P(z_1) \cdot \prod_{j=1} P(z_j, z_{j+1})$$

The algorithm proceeds as follows:

1. Start with a random permutation of the alphabet.
2. Swap two letters at random in the permutation
3. Log-likelihood at the previous and proposed state: $\mathcal{L}_{old}, \mathcal{L}_{new}$
4. Accept the proposed swap with acceptance probability:

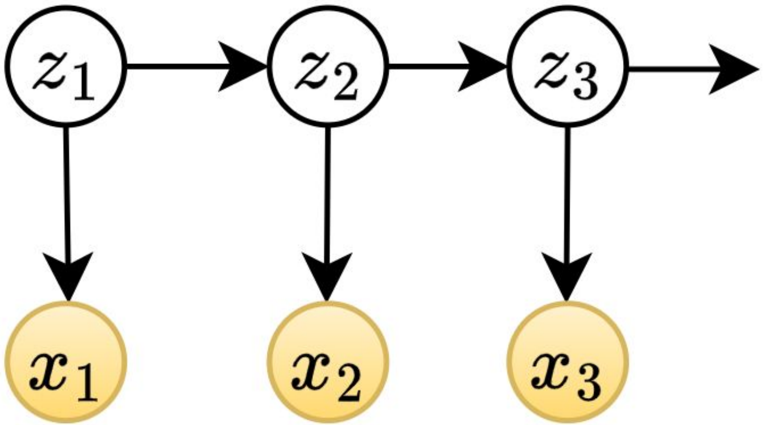
$$\min\{1, \exp(\mathcal{L}_{new} - \mathcal{L}_{old})\}$$



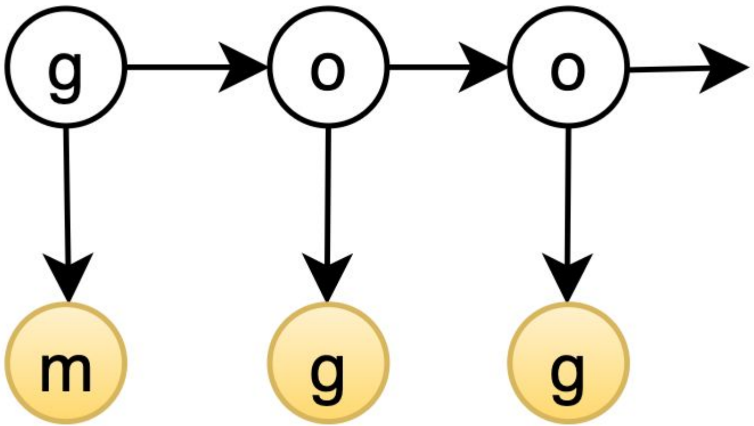
At each step, if the likelihood increases, move to proposed permutation.

This algorithm learns a fixed permutation.

Hidden Markov Models



Hidden Markov Models



$A_{i,j}$: probability of the i-th letter being followed by the j-th letter.

π_i : probability of the chain starting with i-th letter

$B_{i,j}$: probability of the i-th hidden state generating the j-th letter.

1. Obtain $A_{i,j}$, π_i → Learn from available text

2. Estimate $B_{i,j}$ → **Baum-Welch** algorithm (EM) until convergence

3. Obtain the most likely hidden states. → **Viterbi** (Max-Plus)

EM Algorithm

E-step: compute $p(z_n|x,\theta)$ for $n = 1, \ldots, N$ using forward-backward algorithm.

M-step: update $B_{i,j}$ as follows

$$B_{i,j} = P(x_n = j|z_n = i) \propto \sum_{n=1}^{N} \mathbb{1}(x_n = j) P(z_n = i|x)$$

Numerical issues

The issue arises because the messages are computed using the recursion:

$$\alpha(z_n) = p(x_n|z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n|z_{n-1})$$

Typically this probabilities are small number, hence $\alpha(z_n)$ goes quickly to 0, leading to underflow issues.

To tackle this problem, ***scaling factors*** are introduced to keep in the order of unity the messages.

Note that $\alpha(z_n)$ is:

$$\alpha(z_n) = \tilde{p}(x_1, \ldots, x_n, z_n)$$

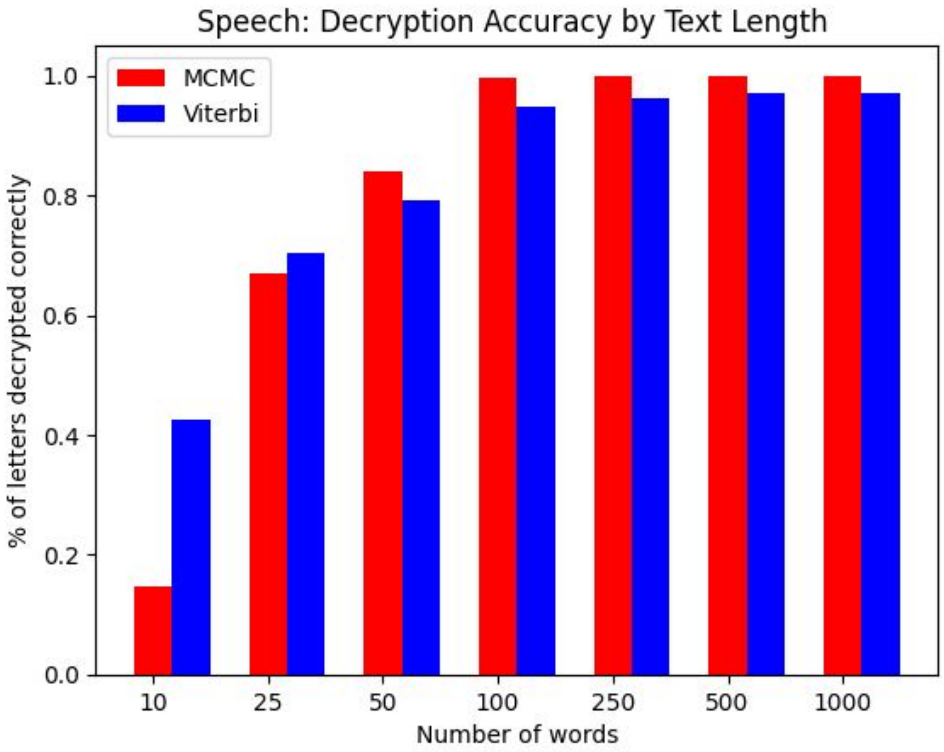
Define a normalized version of the forward messages given by:

$$\hat{\alpha}(z_n) = \frac{\alpha(z_n)}{p(x_1, \ldots, x_n)}$$

Then, scaling factors are defined to relate the scaled and original variables to rescale the backward messages as well:

$$c_n = p(x_n | x_1, \ldots, x_{n-1})$$

Results on substitution cipher



On simple substitution cipher MCMC outperforms HMM.

Homophonic cipher

Number of words	10	25	50	100	250	500	1000
% correct letters	*42.6*	*70.4*	*79.1*	*95.3*	*96.6*	*97.2*	*97.3*

```
twntqw nz lwjewhf w3hn9w k qkf25f0 lkj xkdw emaj
xnhf5fy nf emw unkje nz 6hkf4w c8 ehnn9j
```

pexple of western qurope a landing was mave this morning on the coust of france by troous

Further work

- More in depth comparison of efficiency.
- Could try out on more complex homophonic ciphers [2].
- *Can we exploit linguistic similarities between languages?*

gwhkla mfs cfjs
hggdkgrfk gwhkla mfs
cfjs hggdkgrfk
gwhkla mfs cfjs
hggdkgrfk gwhkla mfs
cfjs hggdkgrfk
gwhkla mfs cfjs
hggdkgrfk

References and links

Referenced Papers

[1] Diaconis, Persi. (2009). The Markov Chain Monte Carlo Revolution. *Bulletin of the American Mathematical Society. 46*

[2] Berg-Kirkpatrick, T., and D. Klein. 2013. Decipherment with a million random restarts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, 18–21 October, Seattle, Washington, 874–878.*

Github repository

https://github.com/alessimichele/HMM-for-text-decryption